

Consultative Committee for Space Data Systems

DRAFT RECOMMENDATION FOR SPACE
DATA SYSTEM STANDARDS

CCSDS FILE DELIVERY PROTOCOL (CFDP)

CCSDS 727.0-BP-1.1

~~BLUE BOOK~~ PINK SHEETS

~~January~~ April 2002



CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION.....	1-1
1.1 PURPOSE AND SCOPE.....	1-1
1.2 APPLICABILITY.....	1-1
1.3 CONVENTIONS AND DEFINITIONS.....	1-1
1.4 REFERENCES	1-5
2 OVERVIEW	2-1
2.1 GENERAL.....	2-1
2.2 ARCHITECTURE ELEMENTS	2-2
2.3 GENERAL CHARACTERISTICS	2-4
2.4 OVERVIEW OF INTERACTIONS.....	2-8
3 SERVICE DESCRIPTION	3-1
3.1 SERVICES AT THE USER INTERFACE	3-1
3.2 SUMMARY OF PRIMITIVES	3-1
3.3 SUMMARY OF PARAMETERS	3-2
3.4 SERVICES REQUIRED OF THE UNDERLYING COMMUNICATION SYSTEM.....	3-6
3.5 CFDP SERVICE PRIMITIVES	3-8
4 PROTOCOL SPECIFICATION.....	4-1
4.1 CORE PROCEDURES.....	4-1
4.2 EXTENDED PROCEDURES	4-21
5 PDU FORMATS	5-1
5.1 GENERAL.....	5-1
5.2 FILE DIRECTIVE PDUs	5-4
5.3 FILE DATA PDU	5-12
5.4 TLV PARAMETERS	5-12
6 USER OPERATIONS	6-1
6.1 RESERVED CFDP MESSAGE FORMAT.....	6-1
6.2 PROXY OPERATION	6-3
6.3 DIRECTORY OPERATIONS.....	6-9
6.4 REMOTE STATUS REPORT OPERATIONS.....	6-11
6.5 REMOTE SUSPEND OPERATIONS	6-14

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
6.6 REMOTE RESUME OPERATIONS	6-17
6.7 STORE AND FORWARD OVERLAY OPERATIONS	6-19
7 CFDP SERVICE CLASSES	7-1
7.1 DEFINED CLASSES	7-1
7.2 FUNCTIONS OF CLASS 1—UNRELIABLE TRANSFER	7-2
7.3 FUNCTIONS OF CLASS 2—RELIABLE TRANSFER	7-5
7.4 FUNCTIONS OF CLASS 3—UNRELIABLE TRANSFER VIA ONE WAYPOINT	7-8
7.5 FUNCTIONS OF CLASS 4—RELIABLE TRANSFER VIA ONE WAYPOINT	7-11
8 MANAGEMENT INFORMATION BASE	8-1
8.1 GENERAL	8-1
8.2 LOCAL ENTITY CONFIGURATION INFORMATION	8-1
8.3 REMOTE ENTITY CONFIGURATION INFORMATION	8-2
ANNEX A INFORMATIVE REFERENCES	A-1
ANNEX B ACRONYMS	B-1
ANNEX C EXAMPLE OF CHECKSUM CALCULATION	C-1
ANNEX D EXTENDED PROCEDURES	D-1

Figure

1-1 Bit Numbering Convention	1-2
1-2 Octet Convention	1-2
2-1 Architectural Elements of the File Delivery Protocol	2-2
2-2 CFDP Procedures	2-5
2-3 Copy Operations, Sequence of Events	2-9
2-4 Put Operations, Sequence of Events	2-10
2-5 File Custody Transfer Operations, Sequence of Events	2-10
2-6 Proxy Put Operations, Sequence of Events	2-11
6-1 SFO Operations	6-20

CONTENTS (continued)

<u>Table</u>	<u>Page</u>
6-23 Remote Resume Request Message	6-18
6-24 Remote Resume Response Message	6-19
6-25 SFO Message Types	6-21
6-26 SFO Request Message	6-22
6-27 SFO Message to User Message	6-23
6-28 SFO Filestore Request Message	6-24
6-29 SFO Fault Handler Override Message	6-24
6-30 SFO Flow Label Message	6-24
6-31 SFO Report Message	6-27
7-1 Class 1 Source Procedures	7-2
7-2 Class 1 Destination Procedures	7-3
7-3 Class 2 Source Procedures	7-5
7-4 Class 2 Destination Procedures	7-6
7-5 Class 3 Source Procedures	7-8
7-6 Class 3 Waypoint Procedures	7-9
7-7 Class 3 Destination Procedures	7-9
7-8 Class 4 Source Procedures	7-11
7-9 Class 4 Waypoint Procedures	7-12
7-10 Class 4 Destination Procedures	7-13
8-1 Local Entity Configuration Information	8-1
8-2 Remote Entity Configuration Information	8-2

~~— Annex D documents proposed ‘extended’ procedures for conducting successive CFDP file copy operations among three or more entities in a store and forward manner.~~

1.3.3 DEFINITIONS

1.3.3.1 Definitions from OSI Basic Reference Model

This Recommendation makes use of a number of terms defined in reference [5]. The use of those terms in this Recommendation shall be understood in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- entity;
- Protocol Data Unit (PDU);
- service;
- Service Access Point (SAP);
- Service Data Unit (SDU).

1.3.3.2 Definitions from Open Systems Interconnection (OSI) Service Definition Conventions

This Recommendation makes use of a number of terms defined in reference [6]. The use of those terms in this Recommendation shall be understood in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- Indication;
- Primitive;
- Request;
- Response.

1.3.3.3 Terms Defined in This Recommendation

Within the context of this document the following definitions apply:

A *CFDP protocol entity* (or *CFDP entity*) is a functioning instance of an implementation of the CFDP protocol, roughly analogous to an Internet protocol ‘host’.

The functional concatenation of a file and related *metadata* transmitted between two CFDP entities is termed a *File Delivery Unit* (FDU); in this context the term ‘metadata’ is used to

refer to any data exchanged between CFDP protocol entities in addition to file content, typically either additional application data (such as a ‘message to user’) or data that aid the recipient entity in effectively utilizing the file (such as file name). Note that an FDU may consist of metadata only. Note also that the term ‘file’ is frequently used in this specification as an abbreviation for ‘file delivery unit’; only when the context clearly indicates that only actual files are being discussed, for example, in the explanation of the segmentation control parameter or the source and destination file name parameters of the CFDP Service Definition, should the term ‘file’ not be read as ‘file delivery unit’.

The individual, bounded, self-identifying items of CFDP data transmitted between CFDP entities are termed *CFDP Protocol Data Units* or *CFDP PDUs*. Unless otherwise noted, in this document the term ‘PDU’ always means ‘CFDP PDU’. CFDP PDUs are of two general types: *File Data PDUs* convey the contents of the files being delivered, while *File Directive PDUs* convey only metadata and other non-file information that advances the operation of the protocol.

A *transaction* is the end-to-end transmission of a single FDU between two CFDP entities. A single transaction normally entails the transmission and reception of multiple PDUs. Each transaction is identified by a unique transaction ID; all elements of any single FDU, both file content and metadata, are tagged with the same CFDP transaction ID.

Any single end-to-end file transmission task has two associated entities: the entity that has the file at the beginning of the task (the *source*) and the entity that has a copy of the file when the task is completed (the *destination*).

Each end-to-end file transmission task comprises one or more point-to-point file copy operations. ~~Each end-to-end file transmission task comprises a single point-to-point file copy operation.~~ A file copy operation has two associated entities: the entity that has a copy of the file at the beginning of the operation (the *sender* or *sending entity*) and the entity that has a copy of the file when the operation is completed (the *receiver* or *receiving entity*). In the simplest case, the only sender of the file is the source and the only receiver is the destination. In more complex cases (the general case) there are additional ‘waypoint’ entities that receive and send copies of the file; the source is the first sender and the destination is the last receiver. ~~(Note that file copy operations are functionally equivalent to file transmission tasks. The distinction between the two is preserved in this document solely as a terminological aid for possible future extension of the protocol to multi-point cases, in which a single file transmission task might comprise multiple successive file copy operations.)~~

Filestore is a generic term referring to the medium used to store files.

The term *offset* is used in a familiar way: the offset of a given octet of file data is the number of data octets that precede this octet in the file.

The file delivery *progress* represented by a given file data PDU is the sum of the offset of the PDU’s file data content (the offset of the content’s first octet) and the length of that file data content.

2 OVERVIEW

2.1 GENERAL

This Recommendation defines a protocol suitable for the transmission of files to and from spacecraft data storage. In addition to the purely file delivery related functions, the protocol also includes file management services to allow control over the storage medium.

The protocol is capable of operating in a wide variety of mission configurations, from relatively simple low earth orbit spacecraft to complex arrangements of orbiters and landers supported by multiple ground facilities and transmission links. In its simplest form, the protocol provides a *Core* file delivery capability operating across a single link. For more complex mission scenarios, the protocol offers *Extended* operation providing store-and-forward functionality across an arbitrary network containing multiple links with disparate availability.

The protocol is independent of the technology used to implement data storage and requires only a few fundamental filestore capabilities in order to operate. It assumes two filestores, one within the spacecraft and one on the ground, and operates by copying data between the two filestore locations.

The protocol makes no assumptions about the information being transferred and can be utilized for a wide range of applications involving the loading, dumping, and control of spacecraft storage.

The protocol has been specifically designed to minimize the resources required for operation. It is also scaleable, so that only those elements required to fulfill the selected options are required to be implemented.

Although the protocol can operate over a wide range of underlying communication services, this Recommendation assumes the use of CCSDS packet services as defined in references [1] through [4].

2.2.2 USER

The protocol operates at the request of the CFDP user. The user interacts with the protocol using the service primitives defined in section 3.

A CFDP user is always a software task, which may or may not be operated by a human.

~~Each CFDP protocol entity has at most one user. In some instances a user may not be present; in particular, any entity which always functions solely as an Extended Procedures waypoint (performing store and forward operations) need not have a user. Each CFDP protocol entity has one user.~~

2.2.3 PROTOCOL ENTITY

~~The protocol entity consists of implementations of the *Core delivery procedures*, which allow immediate file delivery and manipulation over a single network hop, and optionally the *Extended procedures*, which allow for time-disjunct or immediate delivery over a number of network hops with appropriate facilities for onward routing. A single service interface is presented to the user; the operation of the Extended procedures is evident in the quality of service and the multi-hop capability. The protocol entity consists of implementations of the *delivery procedures*, which allow immediate file delivery and manipulation over a single network hop. A single service interface is presented to the user.~~

2.2.4 FILESTORE

The protocol operates by copying files from storage medium to storage medium, and it is therefore assumed that all CFDP entities have access to a local storage capability. As the ways in which the storage capability is provided will vary, the protocol is built on the premise that any file or organized set of files (i.e., a filestore) can be described in terms of a single standard representation. This representation, called a ‘**virtual filestore**’ is assigned a standard set of attributes which are then used by the protocol to manage the file delivery process. In an implementation, the virtual filestore must be mapped to and from actual hardware and software which constitute the real filestore. This approach allows complete independence from the technology used to implement the filestore.

2.2.5 UNDERLYING COMMUNICATION SYSTEM

The protocol assumes the availability of a single conceptual underlying communication system, referred to as the ‘**Unitdata Transfer (UT) layer**’, to which all CFDP entities in a given CFDP addressing domain have access. In order that the protocol may operate over a wide range of implementations, the services required of the UT layer have intentionally been kept as simple as possible; those services are assumed to be made available to any single CFDP entity at only a single conceptual service access point.

Because of the potential diversity of physical underlying services in use, no reuse of underlying protocol features is utilized beyond the addressing required to identify localities

As depicted in figure 2-2, the protocol procedures constitute the interaction between two protocol entities. The sending entity is the entity from which the file is copied in a file copy operation. The receiving entity is the entity to which the file is copied in a file copy operation. Where direct connectivity between the source and destination is impossible, the Extended procedures automatically execute multiple file copy operations: one file copy operation between the source and the first waypoint; others between successive waypoints as necessary; and a final file copy operation between the last waypoint and the destination. Each of these is simply another instance of the Core file copy operation.

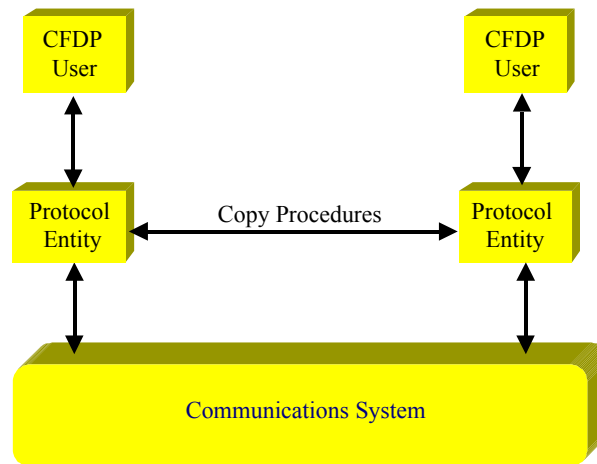


Figure 2-2: CFDP Procedures

The reliability of a transaction is determined by whether the transaction is chosen to operate in unacknowledged mode or one of the acknowledged modes. In unacknowledged mode, data delivery failures are not reported to the sender and therefore cannot be repaired. Reception of the complete file is therefore not guaranteed. In acknowledged mode, the receiver informs the sender of any undelivered file segments or ancillary data. These are then retransmitted, guaranteeing complete file delivery.

Each transaction results in the copying of a single FDU from source to destination. Within any single transaction, the subsidiary actions cancel, suspend, resume, and report may be performed.

The transaction is terminated when one of the following conditions applies:

- the file has been successfully transferred;
- a sending or receiving entity has disallowed the transaction;
- the transaction has been canceled because a fault was detected;
- the source or destination CFDP user has canceled the transaction.

2.3.2 USER OPERATIONS

The term ‘user operations’ refers to the use of the CFDP services offered by the local CFDP entity to cause the CFDP user of a remote CFDP entity to initiate additional CFDP transactions. User operations are implemented using the ‘Message to User’ capability of the protocol to forward an ‘order’ to the remote CFDP user which will in turn initiate a transaction with its local CFDP entity.

Five standard user operations are defined: proxy operations, remote status report operations, remote suspend and resume operations, and directory operations. Directory operations are used to request a listing of the contents of a specified directory in the remote user’s local filestore. Remote status report operations are used to request a report of the status of a specified CFDP transaction at the remote entity. Remote suspend and resume operations are used to request the suspension and resumption of a specified CFDP transaction at the remote entity. Proxy operations are used to initiate the delivery of a file from a remote CFDP entity to some other user, either to the local user itself (in which case the proxy operation functions as a ‘Get’) or to the user of some third CFDP entity. The FDU transmitted in a proxy operation normally contains a file but may contain only metadata, such as filestore directives or a Message to User containing an order to another remote CFDP user.

User operations are described in section 6.

2.3.3 ADDRESSING

Within any single CFDP network (i.e., any set of CFDP entities that is closed under communication), each CFDP entity is assigned a unique name. As each CFDP entity has access to exactly one filestore, identification of a CFDP entity implicitly identifies the associated filestore. At each CFDP entity location, address look-up capabilities are provided using information contained in the associated MIB. This look-up capability provides translation between the network-unique name of a CFDP entity and the corresponding UT address, which may in reality be an Internet address, radio device buffer, APID, virtual channel number, or other implementation-specific mechanism.

The names, or ‘entity IDs’, that uniquely identify the source CFDP entity and the destination CFDP entity must be included in the header of each PDU in order for an FDU to be successfully transmitted through a series of waypoint CFDP entities.

2.3.4 STORE AND FORWARD CONSIDERATIONS

When the Extended procedures are operating, the sender and/or receiver of a given PDU may be a ‘waypoint’ CFDP entity. Extended procedures are used when the original source of the PDU has no direct connectivity to the PDU’s final destination, but only to some intermediate entity. That waypoint entity in turn may have direct connectivity either to the PDU’s final destination or only to some further intermediate entity; the last waypoint entity in such a chain must have direct connectivity to the final destination of the PDU.

The determination of how and when a waypoint entity forwards a PDU toward its target entity is an implementation matter. In general it is desirable to forward each PDU as soon as possible, rather than wait until custody of an entire FDU has been taken before forwarding any part of it; this approach minimizes the time required for complete end-to-end transmission of the data. In practice, however, immediate forwarding will frequently be impossible, because radio contact among CFDP entities is typically discontinuous. The waypoint entity in such cases must store PDUs in some persistent medium, such as an intermediate copy of the transmitted file, until forwarding is practical.

The end-to-end execution of a transaction may therefore comprise multiple successive executions of Core procedures between adjacent entities, some of which may be initiated by the Extended procedures themselves rather than by a CFDP user's invocation of services; when this is the case, the Extended procedures essentially take on the role of the CFDP user. However, this variation is invisible to the Core procedures, which operate in the same way at all times.

When the sender for a Copy File procedure is the file's source entity but the receiver is a waypoint rather than the file's destination entity, the sender issues a Transfer-Consigned.indication primitive (rather than a Transaction-Finished.indication primitive) when it receives notification from the receiver that the file has been successfully copied. This notifies the sending application that custody of the file has been transferred to the first waypoint; if the original file itself is being used as the sender's retransmission buffer, and is therefore protected from deletion or modification, it is now safe to end that protection.

As noted earlier, when the receiver for a Copy File procedure is a waypoint, it may or may not wait until the entire procedure is complete before beginning to copy the file to the destination (or next waypoint). Immediate (incremental) forwarding of the file has the desirable effect of minimizing delay in getting at least part of the file to the destination.

The precise order in which stored PDUs are forwarded when connectivity is established is another implementation matter. Optional 'flow labels' may be associated with transmitted data to aid implementations of CFDP in determining how and when to issue the data.

2.3.5 RETRANSMISSION STRATEGIES

The quality of service offered by the protocol is selectable, according to mission requirements and transmission capability, and ranges from an unacknowledged option, whereby a file is transmitted with no attempt at completeness should errors occur (errors will be detected and data discarded), to a fully acknowledged option providing error recovery through retransmission. For the acknowledged modes of operation, several sub-options may be selected by the receiver. These sub-options relate to release time of any Negative Acknowledgments (NAKs) and range from immediate release to deferred release (whereby any NAKs are stored until the assumed end of the transmission). The extent of available retransmission strategies is more completely described in section 4.

2.3.6 VIRTUAL FILESTORE

The virtual filestore concept provides for a mapping of protocol filestore directives to actual filestore manipulation. The way in which this mapping is performed in an implementation is a local matter. Allowances are made in the protocol for inclusion of additional filestore directives using the Type-Length-Value (TLV) capability of the Metadata PDU.

To enable interoperability, this Recommendation assumes a minimum set of capabilities from the virtual filestore as follows:

- create file;
- delete file;
- rename file;
- append file;
- replace file;
- create directory;
- remove directory;
- list directory.

In some circumstances, it is advantageous for the CFDP protocol to be able to recognize record boundaries within the file. If this option is to be used, the filestore must have the capability to make the distinction between such files and those which are to be treated as a stream of octets.

2.4 OVERVIEW OF INTERACTIONS

Figure 2-3 shows the details of a Copy operation. (In all following figures in this section, dashed lines denote optional actions.)

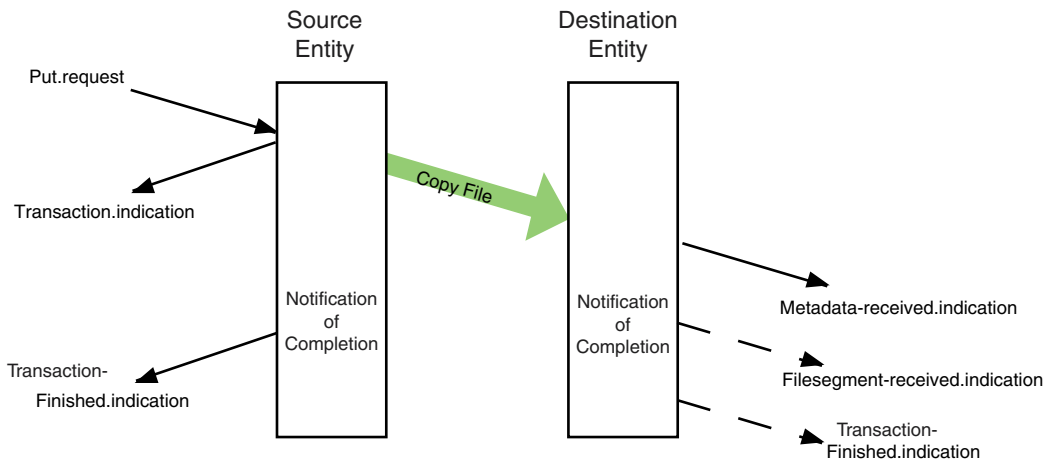


Figure 2-4: Put Operations, Sequence of Events

It is possible for the source and destination CFDP entities of a transaction to be unable to communicate directly; in this case the transaction may entail a series of point-to-point (sender-to-receiver) PDU exchange sessions between the source and destination CFDP entities and one or more waypoint CFDP entities, as shown in figure 2-5.

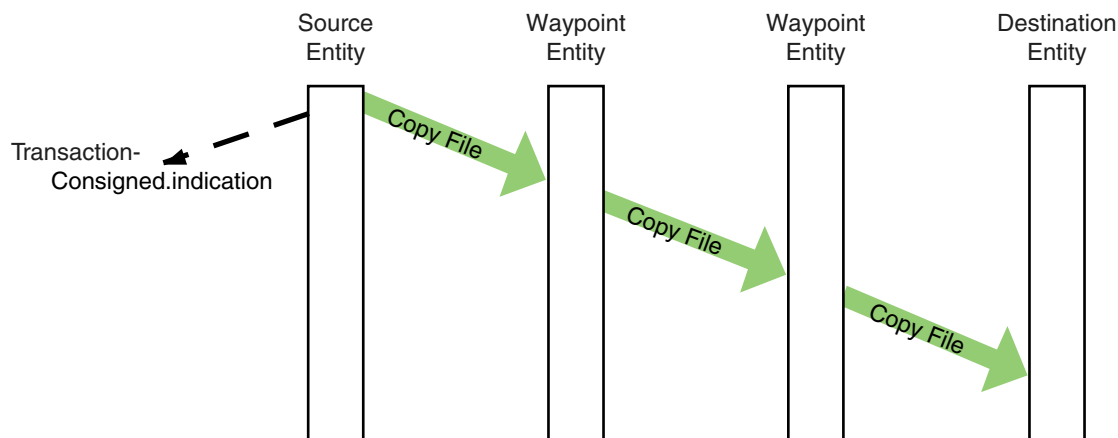


Figure 2-5: File Custody Transfer Operations, Sequence of Events

When the source of the file to be transferred is other than the local filestore, the requesting user must initiate a Put transaction containing a message to the user of the remote CFDP entity, the 'proxy user'. This message requests that the proxy user in turn submit a Put.request to that entity. The destination of the requested Put transaction may be the original requesting user (in which case the proxy operation functions as a 'Get') or some third CFDP user. On completion of the second transaction, the proxy user initiates a third transaction to notify the original requesting user of the completion of the proxy operation. A proxy put operation is illustrated in figure 2-6.

3.2.2 The CFDP service shall deliver the following indication primitives:

- a) Transaction.indication;
- b) EOF-Sent.indication;
- c) Transaction-Finished.indication;
- d) Metadata-Recv.indication;
- e) File-Segment-Recv.indication;
- f) Report.indication;
- g) Suspended.indication;
- h) Resumed.indication;
- i) Fault.indication;
- j) Abandoned.indication.

3.2.3 When the Extended procedures are implemented, the CFDP service shall also deliver the following indication primitive: Transfer-Consigned.indication.

3.3 SUMMARY OF PARAMETERS

NOTE – The availability and use of parameters for each primitive are indicated in 3.5, where parameters that are optional are identified with square brackets [thus]. The following definitions apply.

3.3.1 The *destination CFDP entity ID* parameter shall uniquely identify the CFDP entity **to** which the FDU is to be sent.

3.3.2 The *source CFDP entity ID* parameter shall uniquely identify the CFDP entity **from** which the FDU is to be sent.

3.3.3 The *source file name* parameter

- a) shall contain the full path name at which the file to be copied is located at the filestore associated with the source entity;
- b) shall be omitted when the FDU to be Put contains only metadata, such as a message to a user or a standalone filestore request.

3.3.4 The *destination file name* parameter

- a) shall contain the full path name to which the file to be copied will be placed at the filestore associated with the destination entity;

3.5.16 Transfer-Consigned.indication

3.5.16.1 Function

When (and only when) the Extended procedures are in operation, the Transfer-Consigned.indication primitive shall be used to indicate, to the source CFDP user, the complete reception of a file by the first waypoint.

3.5.16.2 Semantics

Transfer-Consigned.indication shall provide parameters as follows:

Transfer-Consigned.indication (transaction ID)

3.5.16.3 When Generated

Transfer-Consigned.indication shall be generated by the source entity on receipt of a Finished PDU with End System Status flag set to '0'.

3.5.16.4 Effect on Receipt

Transfer-Consigned.indication shall authorize the source CFDP user to modify or delete any retransmission buffer (file) that it was sharing with the protocol entity to conserve persistent storage space.

3.5.16.5 Additional Comments

The Transfer-Consigned.indication primitive is provided to support 'buffer sharing' in implementations of the extended procedures for memory-constrained environments.

4 PROTOCOL SPECIFICATION

4.1 **CORE PROCEDURES**

4.1.1 CRC PROCEDURES

4.1.1.1 CRC Procedures at the PDU Transmitting Entity

If the CRC option is active, the PDU transmitting entity shall set the CRC flag to ‘true’ and calculate and insert the CRC for each outgoing PDU.

4.1.1.2 CRC Procedures at the PDU Receiving Entity

If the CRC flag is set to ‘true’ in the incoming PDU, the PDU receiving entity shall calculate the CRC and discard the PDU if it fails the CRC validation procedure.

4.1.1.3 CRC Validation Procedure

4.1.1.3.1 The CRC computation algorithm shall be the standard CCSDS Telecommand CRC algorithm as specified in 4.2.1.3 of the CCSDS Telecommand Recommendation (reference [4]).

4.1.1.3.2 The CRC value shall be placed in the final octets of the PDU data field, and its length shall be included in the PDU data field length. The CRC algorithm shall be applied from the first octet of the PDU header to the last octet of the PDU data field prior to the CRC value.

4.1.2 CHECKSUM PROCEDURES

The checksum shall be 32 bits in length and shall be calculated by the following method (see annex C for an example):

- a) it shall initially be set to all ‘zeroes’;
- b) it shall be calculated by modulo 2^{32} addition of all 4-octet words, aligned from the start of the file;
- c) each 4-octet word shall be constructed by copying some octet of file data, whose offset within the file is an integral multiple of 4 (such as 0, 4, 8, 12, etc.), into the first (high-order) octet of the word, and copying the next three octets of file data into the next three octets of the word;
- d) the results of the addition shall be carried into each available octet of the checksum unless the addition overflows the checksum length, in which case carry shall be discarded.

4.1.5 PDU FORWARDING PROCEDURES

4.1.5.1 Each outgoing PDU shall be directed to the appropriate CFDP entity using the addressing capabilities of the underlying Unitdata Transfer service.

4.1.5.2 The remote Unitdata Transfer Service Access Point (UT-SAP) to which the PDU is to be directed shall be obtained from the MIB, based on the identifier of the CFDP entity to which the PDU is to be delivered.

4.1.5.3 Depending on the type of PDU, the identifier of the CFDP entity to which the PDU is to be delivered shall be one of the following:

- a) for an ACK PDU, the ID of the CFDP entity which sent the PDU that is being acknowledged;
- b) for a File Data, Metadata, EOF, or Prompt PDU, the receiver entity ID for that file copy operation;
- c) for a NAK, Finished, or Keep Alive PDU, the sender entity ID for that file copy operation.

NOTE – When only the Core procedures are implemented, the sender and receiver of the file are the source and destination entities as identified in the PDU header. When the Extended procedures are operating the senders and receivers of files must recognize themselves as such; the manner in which this is accomplished is an implementation matter.

4.1.5.4 Any PDU whose destination entity ID differs from the identifier of the CFDP entity to which it is delivered shall be discarded, except that such PDUs ~~may~~ **shall** be processed in accordance with the ~~proposed~~ Extended Procedures if those procedures are implemented.

4.1.6 COPY FILE PROCEDURES

NOTES

- 1 In the following subsections, the name of an EOF PDU or a Finished PDU includes, in parentheses, either the name of the specific condition code that the PDU carries (such as ‘No error’) or the general condition name ‘cancel’ indicating that the PDU carries some condition code other than ‘No error’ and ‘Suspend request received’.
- 2 The name of an ACK PDU includes, in parentheses, the type of PDU which is being acknowledged: EOF or Finished.
- 3 When only the Core procedures are implemented, the sender of the file is always the transaction’s source entity, and the receiver of the file is always the transaction’s destination entity.

4.1.6.4.1.4 The procedures for processing a Finished (No error) PDU with End System Status flag set to '0' are defined in the Extended procedures.

4.1.6.4.1.5 Positive Acknowledgment procedures shall be applied to the EOF (No error) and Finished (No error) PDUs with the Expected Responses being ACK (EOF) PDU and ACK (Finished) PDU, respectively.

4.1.6.4.2 Acknowledged Mode Procedures at the Receiving Entity

4.1.6.4.2.1 A file data gap shall be detected when any of the following conditions is true:

- a) the offset of the first octet of the first extent of received file data is not zero;
- b) the offset of the last octet of some contiguous extent of file data differs from the offset of the first octet of the next contiguous extent of data of the same file by more than one;
- c) the offset of the last octet of the last extent of file data differs from the length of the file as stated in the EOF (No error) PDU by more than one.

4.1.6.4.2.2 Lost Metadata shall be detected upon receipt of the first non-Metadata PDU (File Data or EOF (No error)) of a transaction for which no Metadata PDU has yet been received.

4.1.6.4.2.3 In accordance with any Lost Segment Detection procedures that are in effect, the receiving CFDP entity shall issue NAK PDUs as follows:

- a) Each NAK PDU shall identify the subset of file data to which it pertains, i.e., the *scope* of the NAK PDU. The scope of a NAK PDU is expressed as two offsets within the file, indicating the start and end of the scope.
- b) In addition to its scope, each NAK PDU shall contain zero or more segment requests. The segment request(s) in a NAK PDU shall identify the start offsets and end offsets of all extents of file data within its scope which have not yet been received, and shall also identify missing metadata if any.
- c) The start offsets and end offsets of segment requests do not need to relate in any way to the original segmentation of the file.
- d) One *NAK sequence* shall be issued for each event upon which gap reporting is deemed necessary by any Lost Segment Detection procedures that are in effect.
 - 1) Each NAK sequence shall pertain to some subset of file data, i.e., the *scope* of the NAK sequence.
 - 2) The start of scope for a NAK sequence shall be the end of scope of the previous NAK sequence issued for the same transaction, or zero if there has been no prior NAK sequence.

4.1.11 INTERNAL PROCEDURES

4.1.11.1 Notice of Completion Procedures

NOTES

- 1 In the following subsections, the name of a Notice of Completion may include, in parentheses, the name of the specific disposition of the affected Copy File procedure, either 'Completed' indicating successful completion of the procedure or 'Canceled' indicating cancellation of the procedure. Disposition name may be omitted from the name of a Notice of Completion when it is irrelevant in the context in which the name is used.
- 2 It is possible for a Notice of Completion (Canceled) to be issued after a Notice of Completion (Completed) has been issued for the same transaction. This could be caused, for example, by the receiving entity's repeated failure to receive an ACK PDU in response to its Finished (No error) PDU, resulting in the declaration of a Positive ACK Limit Reached fault. The standard Notice of Completion Procedures continue to apply in this event. At the receiving entity the received file will already have been delivered to the user for normal processing by the time the Notice of Completion (Canceled) is issued.

4.1.11.1.1 Notice of Completion Procedures at the Sending Entity

4.1.11.1.1.1 On Notice of Completion of the Copy File procedure, the sending CFDP entity shall

- a) release all unreleased portions of the file retransmission buffer;
- b) stop transmission of file segments and metadata.

4.1.11.1.1.2 If sending in acknowledged mode,

- a) any transmission of Prompt PDUs shall be terminated;
- b) the application of Positive Acknowledgment Procedures to PDUs previously issued by this entity shall be terminated.

4.1.11.1.1.3 In any case,

- a) if the sending entity is the transaction's source, it ~~the CFDP entity~~ shall issue a Transaction-Finished.indication primitive indicating the condition in which the transaction was completed;
- b) if the sending entity is the transaction's source and the file was sent in acknowledged mode and the procedure disposition cited in the Notice of Completion is 'Completed' and the Finished PDU whose arrival completed the transaction contained a Filestore

Responses parameter, then that Filestore Responses parameter shall be passed in the Transaction-Finished.indication primitive.

4.1.11.1.2 Notice of Completion Procedures at the Receiving Entity

4.1.11.1.2.1 If receiving in acknowledged mode:

- a) transmission of NAK PDUs, whether in response to NAK timer expiration or in response to any other events, shall be terminated;
- b) any transmission of Keep Alive PDUs shall be terminated;
- c) the application of Positive Acknowledgment Procedures to PDUs previously issued by this entity shall be terminated.

4.1.11.1.2.2 In any case,

- a) if the receiving entity is the transaction's destination and the procedure disposition cited in the Notice of Completion is 'Completed', the receiving CFDP entity shall execute any filestore requests conveyed by the Put procedure;
- b) if the procedure disposition cited in the Notice of Completion is 'Canceled', and the receiving entity is the transaction's destination, then the incomplete data shall be either discarded or retained according to the option set in the MIB;

NOTE – On Notice of Completion (Canceled) for a transaction for which a Notice of Completion (Completed) was previously declared, all file data have necessarily already been received and, therefore, there are no incomplete data to discard or retain.

- c) if the receiving entity is the transaction's destination, then it ~~the CFDP entity~~ may optionally issue a Transaction-Finished.indication primitive indicating the condition in which the transaction was completed;
- d) if the receiving entity is the transaction's destination, Filestore Responses and/or a Status Report shall be passed as parameters of the Transaction-Finished.indication primitive as available.

4.1.11.2 Notice of Cancellation Procedures

4.1.11.2.1 General

At any time during a Copy File procedure, either the sending CFDP entity or receiving CFDP entity may issue a Notice of Cancellation.

NOTE – The Notice of Cancellation may be issued in reaction to the declaration of a fault or to receipt of a Cancel.request primitive submitted by the CFDP user.

4.1.11.3 Notice of Suspension Procedures

4.1.11.3.1 General

4.1.11.3.1.1 At any time during a Copy File procedure, either the sending CFDP entity or the receiving CFDP entity may issue a Notice of Suspension.

NOTE – A Notice of Suspension may be issued in reaction to the declaration of a fault or to receipt of a Suspend.request primitive submitted by the CFDP user.

4.1.11.3.1.2 However, a Notice of Suspension shall be ignored if it pertains to a transaction that is already suspended or if it is issued by the receiving CFDP entity for a transaction sent in Unacknowledged mode.

4.1.11.3.1.3 The following lists of the effects of transaction suspension at the sending and receiving entities are exhaustive; no additional effects should be inferred. In particular, received EOF and Finished PDUs shall be acknowledged and processed in accordance with Positive Acknowledgment Procedures and the relevant Copy File procedures regardless of any transaction suspension that may be in effect.

4.1.11.3.2 Notice of Suspension Procedures at the Sending Entity

4.1.11.3.2.1 On Notice of Suspension of the Copy File procedure, the sending CFDP entity shall

- a) suspend transmission of file segments;
- b) save the status of the transaction.

4.1.11.3.2.2 If operating in acknowledged mode,

- a) any transmission of Prompt PDUs shall be suspended;
- b) the inactivity timer shall be suspended;
- c) the application of Positive Acknowledgment Procedures to PDUs previously issued by this entity shall be suspended.

4.1.11.3.2.3 ~~The sending entity~~ If the sending entity is the transaction's source entity, it shall issue a Suspended.indication.

4.1.11.3.3 Notice of Suspension Procedures at the Receiving Entity

On Notice of Suspension of the Put procedure, the receiving CFDP entity shall

- a) suspend transmission of NAK PDUs;
- b) suspend any transmission of Keep Alive PDUs;

Resumed.indication shall be issued and (b) thawing transmission for a suspended transaction shall have no effects whatsoever.

4.1.12.3 Reception Opportunity Procedures

4.1.12.3.1 On notification of the end of an opportunity to receive from a specified remote CFDP entity, the CFDP entity shall freeze reception for all transactions for which it is the receiving entity and the specified remote entity is the sending entity, except those that are in Unacknowledged mode.

4.1.12.3.2 The freezing of reception for a transaction shall have the same effects as suspension of that transaction by the receiving entity (see 4.1.11.3.3), except that no Suspended.indication shall be issued and the transaction shall not be considered suspended.

4.1.12.3.3 On notification of the start of an opportunity to receive from a specified remote CFDP entity, the CFDP entity shall thaw reception for all transactions for which it is the receiving entity and the specified remote entity is the sending entity, except those that are in Unacknowledged mode.

4.1.12.3.4 The thawing of reception for a transaction shall have the same effects as resumption of that transaction by the receiving entity (see 4.1.6.7.3), except that (a) no Resumed.indication shall be issued and (b) thawing reception for a suspended transaction shall have no effects whatsoever.

4.2 EXTENDED PROCEDURES

4.2.1 GENERAL

4.2.1.1 Because the Extended procedures supplement and rely on the capabilities provided by the Core procedures, a CFDP implementation that supports the Extended procedures shall also support all of the Core procedures without deviation from the specification.

4.2.1.2 In particular, waypoint entities involved in a Copy File procedure shall have the same ability to cancel, suspend, and resume the transaction as do the source and destination entities. To this end, whenever a CFDP entity is functioning as a waypoint (i.e., it is functioning as the receiving entity for a Copy File procedure whose file is destined for some other entity), the Extended procedures shall function in the capacity of the CFDP user: they shall receive all CFDP service indications issued in the course of the Copy File procedure and shall submit any necessary CFDP service requests.

4.2.2 FILE DATA RELAY PROCEDURES

4.2.2.1 General

The following additional procedures shall apply whenever the receiving entity for a Copy File procedure is not the file's destination.

The transmission mode of a PDU relayed by a waypoint shall be the same as the transmission mode of that PDU as received at that waypoint.

4.2.2.2 File Data Relay Procedures at the Receiving Entity in Unacknowledged Mode

Receipt of a Metadata, File Data, or EOF (No error) PDU shall cause the receiving CFDP entity to relay the PDU either to the destination entity or to another waypoint.

4.2.2.3 File Data Relay Procedures at the Sending Entity in Any Acknowledged Mode

On receipt of a Finished (No error) PDU with End System Status flag set to '0', the sending CFDP entity shall issue a Notice of Consignment.

NOTE – The transaction remains unfinished after a Notice of Consignment and therefore remains liable to cancellation.

4.2.2.4 File Data Relay Procedures at the Receiving Entity in Any Acknowledged Mode

4.2.2.4.1 Following receipt of the first PDU for a new transaction, the receiving CFDP entity shall automatically initiate a second Copy File procedure, functioning as the sender.

NOTES

- 1 The receiver of the second Copy File procedure is either the file's destination or another waypoint. The receiving entity shall be selected at the time the procedure is initiated, based on the destination entity ID in the header of the PDU whose reception triggered initiation of the procedure.
- 2 The segmentation algorithm exercised by the waypoint entity in performing this Copy File procedure will not necessarily be the same as that exercised by any prior or subsequent sender of the file. That is, the sizes of File Data PDUs may be different during different Copy File procedures of the same transaction.
- 3 Features of CFDP that specifically rely on in-sequence data arrival should not be utilized in any operational scenario that relies on the Extended Procedures in any acknowledged mode. These features include the Keep Alive mechanism and Incremental Lost Segment Detection. In any acknowledged mode, data retransmission may cause the order in which file data segments are forwarded from a waypoint to differ from the order in which they were originally issued by the source entity.

4.2.2.4.2 Upon receipt of each Metadata PDU or File Data PDU, the recipient entity shall relay the PDU to the receiver of the second Copy File procedure.

4.2.2.4.3 When file delivery is deemed complete (as explained in 4.1.6.1.2.8) at a waypoint CFDP entity, the receiving CFDP entity shall relay the EOF (No error) PDU to the receiver of the second Copy File procedure and shall issue to the sending entity of the original Copy File procedure a Finished (No error) PDU with End System Status flag set to '0'.

4.2.3 FINISHED (NO ERROR) PDU RELAY PROCEDURE

On receipt of any Finished (No error) PDU with End System Flag set to '1' at any CFDP entity that is not the source CFDP entity for the transaction, the following procedures shall apply in addition to the Acknowledged Mode Procedures at the Sending Entity defined in the Core Procedures:

- a) the recipient CFDP entity shall relay the PDU either to the source entity or to another waypoint;
- b) positive Acknowledgement procedures shall apply to the relayed Finished (No error) PDU.

4.2.4 CANCEL PROPAGATION PROCEDURES

4.2.4.1 On receipt of an EOF (cancel) PDU by a CFDP entity that is neither the source nor the destination of the file,

- a) the entity shall issue a Notice of Completion (Canceled);
- b) if an acknowledged mode is in effect, Positive Acknowledgement procedures shall be applied to the EOF (cancel) PDU with the Expected Response being an ACK (EOF) PDU;
- c) the recipient CFDP entity shall relay the EOF (cancel) PDU toward the destination of the file;
- d) if operating in acknowledged mode, Positive Acknowledgment procedures shall be applied to the relayed EOF (cancel) PDU with the Expected Response being an ACK (EOF) PDU.

NOTE – All of the core Notice of Completion (Canceled) procedures shall apply, with the proviso that the recipient CFDP entity shall be considered both the Sending Entity and also the Receiving Entity.

4.2.4.2 On receipt of a Finished (cancel) PDU by a CFDP entity that is neither the source nor the destination of the file,

- a) the entity shall issue a Notice of Completion (Canceled);

- b) Positive Acknowledgement procedures shall be applied to the Finished (cancel) PDU, with the Expected Response being an ACK (Finished) PDU;
- c) the recipient CFDP entity shall relay the Finished (cancel) PDU toward the source of the file;
- d) Positive Acknowledgment procedures shall be applied to the relayed Finished (cancel) PDU with the Expected Response being an ACK (Finished) PDU.

NOTE – All of the core Notice of Completion (Canceled) procedures shall apply, with the proviso that the recipient CFDP entity shall be considered both the Sending Entity and also the Receiving Entity.

4.2.4.3 On receipt of a Resume request primitive by a CFDP entity that is neither the source nor the destination of the file, all of the core Resume Procedures shall apply, with the proviso that the recipient CFDP entity shall be considered both the Sending Entity and also the Receiving entity.

4.2.5 DEFERRED TRANSMISSION PROCEDURE

The CFDP entity shall retain for later transmission any PDU that cannot immediately be forwarded (because of link outage, for example) to the CFDP entity to which it is to be delivered or to the next waypoint, as applicable.

NOTES

- 1 The means by which the CFDP entity determines when PDUs can and cannot immediately be forwarded is an implementation matter. The CFDP entity might, for example, rely on additional service indications from the UT layer.
- 2 The flow label values associated with transactions may be taken as hints regarding the order in which PDUs should be transmitted when the opportunity arises, but the manner in which flow labels are used is strictly an implementation matter.

4.2.6 INTERNAL EXTENDED PROCEDURES

4.2.6.1 Notice of Consignment Procedures

4.2.6.1.1 On Notice of Consignment of the Copy File procedure, the sending CFDP entity shall:

- a) release all unreleased portions of the file retransmission buffer;
- b) stop transmission of file segments and metadata;
- c) terminate any transmission of Prompt PDUs;

- d) terminate the application of Positive Acknowledgment Procedures to PDUs previously issued by this entity.

4.2.6.1.2 If the sending entity is the source entity, the CFDP entity shall issue a Transfer-Consigned indication primitive.

4.2.6.2 Notice of Cancellation Procedures

On Notice of Cancellation of the Copy File procedure at a CFDP entity that is neither the source nor the destination of the file, all of the core Notice of Cancellation Procedures shall apply, with the proviso that the recipient CFDP entity shall be considered both the Sending Entity and also the Receiving entity.

NOTE – The receiving CFDP entity for the resulting EOF (cancel) PDU is the file's destination or another waypoint. The receiving CFDP entity for the resulting Finished (cancel) PDU is the file's source or another waypoint.

4.2.6.3 Notice of Suspension Procedures

On Notice of Suspension of the Copy File procedure at a CFDP entity that is neither the source nor the destination of the file, all of the core Notice of Suspension Procedures shall apply, with the proviso that the recipient CFDP entity shall be considered both the Sending Entity and also the Receiving entity.

Table 5-6: End-of-File PDU Contents

Parameter	Length (bits)	Values	Comments
Condition Code	4	See table 5-5.	
Spare	4		
File Checksum	32		See section 4.1.2, Checksum Procedures.
File size	32		In octets. This value shall be the total number of file data octets transmitted by the sender, regardless of the condition code (i.e., it shall be supplied even if the condition code is other than 'No error').
Fault Location	TLV	An entity ID TLV; see 5.4.6.	Omitted if condition code is 'No error'. Otherwise, entity ID in the TLV is the ID of the entity at which transaction cancellation was initiated.

5.2.3 FINISHED PDU

The contents of the Parameter field for a File Directive having a Code of Finished PDU shall be as shown in table 5-7.

Table 5-7: Finished PDU Contents

Parameter	Length (bits)	Values	Comments
Condition Code	4	See table 5-5.	
End System Status	1	'0' — Generated by Waypoint '1' — Generated by End System	NOTE: a value of '0' in end system status is meaningful if the proposed Extended Procedures are implemented.
Delivery Code	1	'0' — Data Complete '1' — Data Incomplete	'Data Complete' means that Metadata, all file data, and EOF have been received, and the checksum has been verified.

File Status	2	'00' — Delivered file discarded deliberately '01' — Delivered file discarded due to filestore rejection '10' — Delivered file retained in filestore successfully '11' — Delivered file status unreported	File status is meaningful only when the transaction includes the transmission of file data.
Filestore Responses	TLVs	See table 5-17.	A filestore response TLV must be included for each filestore request TLV of the Metadata PDU.
Fault Location	TLV	An entity ID TLV; see 5.4.6.	Omitted if condition code is 'No error'. Otherwise, entity ID in the TLV is the ID of the entity at which transaction cancellation was initiated.

Table 5-8: ACK PDU Contents

Parameter	Length (bits)	Values	Comments
Directive code	4	See table 5-4. Only EOF and Finished PDUs are acknowledged.	Directive code of the acknowledged PDU.
Directive subtype code	4		Values depend on directive code. For ACK of Finished PDU: binary '0000' if generated by waypoint, binary '0001' if generated by end system. (NOTE: this discrimination is meaningful if the proposed Extended Procedures are implemented.) Binary '0000' for ACKs of all other file directives.
Condition code	4	See table 5-5.	Condition code of the acknowledged PDU.
Spare	2		
Transaction status	2		Status of the transaction in the context of the entity that is issuing the acknowledgment.

5.4.6 ENTITY ID TLV

The Type of the Entity ID TLV shall be 06 hex; the Value shall be an Entity ID as discussed in 5.1.

6.6.4.2 Remote Resume Response

The Remote Resume Response message shall be constructed as indicated in table 6-24.

Table 6-24: Remote Resume Response Message

Field	Size (bits)	Values	Comments
Suspension indicator	1	'0' — not suspended '1' — suspended	Because multiple motivations for suspending a transaction may be concurrently valid, the successful processing of a Remote Resume Request may not actually change the suspension status of the affected transaction.
Transaction status	2	Same values as defined for transaction status in the ACK PDU (see table 5-8)	
Spare	5	All 'zeroes'	
Spare	1	'0'	This field and the following five fields identify the transaction whose resumption was requested; they are taken from the resume request.
Length of Entity ID	3		
Spare	1	'0'	
Length of Sequence Number	3		
Source Entity ID	Variable		
Transaction Sequence Number	Variable		

6.7 STORE AND FORWARD OVERLAY OPERATIONS

6.7.1 GENERAL

The CFDP Store and Forward Overlay (SFO) system is an alternative mechanism for transmitting files between users of CFDP entities which may never be in direct communication; this mechanism does not rely on implementation of the Extended Procedures.

Each transmitted file is received, stored, and forwarded in a hop-by-hop manner by intermediate *waypoint users* (rather than intermediate waypoint CFDP entities, as in the Extended Procedures) until it finally reaches a user—termed the *agent*—whose CFDP entity can directly communicate with that of the *destination* user. The file to be transmitted and all associated metadata are encapsulated in an *SFO transmission file delivery unit* for transmission to each waypoint. The SFO transmission file delivery unit metadata additionally contains an ‘SFO Request’ Message to User that identifies and characterizes the transmission.

The agent user transmits the content of the received SFO transmission file delivery unit—the original file, together with the original associated metadata as extracted from the SFO transaction’s own metadata—to the destination user in a conventional single-hop CFDP transaction termed an *SFO final delivery transaction*. This is similar to the operation of the ‘respondent’ in a Proxy file transfer, except that the SFO Request message is delivered to the destination user in addition to all the original metadata. The agent is then responsible for transmitting an *SFO transmission report file delivery unit*—again via waypoint users—back to the user which was the original *source* of the file.

Errors in routing or relaying and (optionally) successful relay operations are likewise reported back to the original source user in SFO transmission report file delivery units. Relaying failures and (optionally) successes are additionally reported to the final destination user where possible.

Figure 6-1 illustrates SFO operations. The thick right arrows represent SFO transmission file delivery units, while the thin right arrow represents the file delivery unit of an SFO final delivery transaction. The left arrows represent SFO transmission report file delivery units, generated either by the agent waypoint to report on final file delivery, or by any waypoint to report on routing or relay errors.

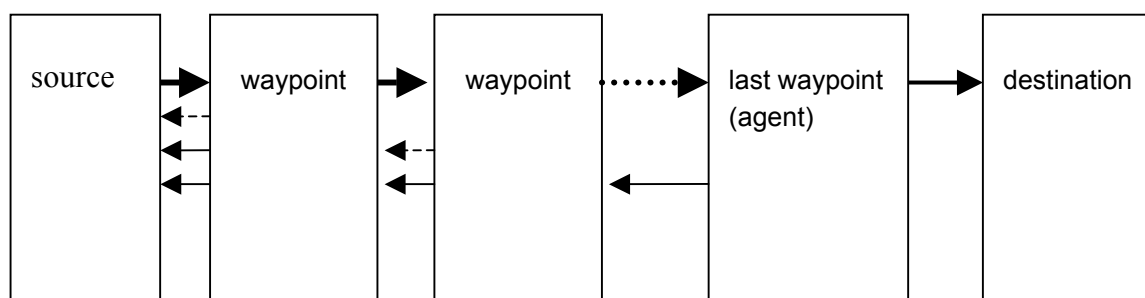


Figure 6-1: SFO Operations

To enable interoperability, mandatory behavior specified in subsections 6.7.2 through 6.7.7 shall be observed by CFDP user applications that are in compliance with the Store and Forward Overlay specification.

6.7.2 SFO MESSAGE TYPES

The message type field for each Reserved CFDP Message used in SFO operations shall contain one of the values specified in table 6-25.

Table 6-25: SFO Message Types

Message Type (hexadecimal)	Interpretation
40	SFO Request
41	SFO Message to User
42	SFO Flow Label
43	SFO Fault Handler Override
44	SFO Filestore Request
45	SFO Report
46	SFO Filestore Response

6.7.3 ROUTING

The SFO Routing procedure shall be as follows:

- The user application shall have access to an information base that associates with each known CFDP entity the IDs of all other entities from which that entity can receive files directly by means of CFDP.
- The user application shall compute from this information base the shortest possible route to the final destination user. A route shall be a list of CFDP entities, beginning with the user application's own local entity and ending with the local entity of the final destination user, such that every entity in the list is able to receive files directly via CFDP from the preceding entity in the list.

6.7.4 INITIATING AN SFO TRANSMISSION

6.7.4.1 General

6.7.4.1.1 In order to initiate SFO transmission of a file and associated metadata, the CFDP user shall invoke the SFO Routing procedure (see 6.7.3) to determine a route from its local CFDP entity to that of the final destination user.

6.7.4.1.2 If the computed route is an empty list, then transmission is impossible. The attempt to initiate SFO transmission shall therefore immediately fail.

6.7.4.1.3 If the computed route contains only the local CFDP entity, then the user application is the final destination of the file. The attempt to initiate SFO transmission shall therefore immediately fail.

6.7.4.1.4 If the computed route contains only the local CFDP entity and that of the final destination user, then the user application shall simply use the CFDP Put.request primitive to request direct delivery of the file and associated metadata to the final destination. No further SFO functionality shall apply.

6.7.4.1.5 If the computed route contains one or more other CFDP entities (waypoints) in addition to the local CFDP entity and the final destination user's CFDP entity, then the user application shall use the CFDP Put.request primitive to request delivery of an SFO transmission file delivery unit to the first waypoint in the route. The file transmitted in the SFO transmission FDU shall be the file that is to be delivered to the final destination user, if any. The metadata of the SFO transmission FDU shall be constructed from the metadata associated with that file and shall comprise a single SFO Request message, defined in 6.7.4.2, together with zero or more of the Reserved CFDP Messages defined in 6.7.4.3 through 6.7.4.6. The destination file name, transmission mode, segmentation control, fault handler overrides, and flow label that constrain the resulting transaction shall be those specified in the metadata of the SFO transmission FDU.

NOTE – At any time after the transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.4.2 SFO Request

The SFO Request message is mandatory and shall be constructed as indicated in table 6-26.

Table 6-26: SFO Request Message

Field	Size (bits)	Values	Comments
Prior waypoints count	5	.	Set to zero by original source user, incremented by every waypoint.
Trace requested	1	1 if true, 0 if false.	
Transmission mode	1	As defined in the CFDP Recommendation.	Specifies the transmission mode for each transmission to or by a waypoint.
Segmentation control	1	As defined in the CFDP Recommendation.	Specifies file segmentation control for each transmission to or by a waypoint.

SFO request label	Variable	LV	A transaction identifier constructed in implementation-specific manner by the original source user. May be used for transaction accounting purposes by the original source user.
Source entity ID	Variable	LV	The ID of the original source user's CFDP entity.
Destination entity ID	Variable	LV	The ID of the final destination user's CFDP entity.
Source file name	Variable	LV	Length is zero if parameter is omitted.
Destination file name	Variable	LV	Length is zero if parameter is omitted.

6.7.4.3 SFO Message to User

One or more SFO Message to User messages may be optionally included and shall be constructed as indicated in table 6-27.

Table 6-27: SFO Message to User Message

Field	Size (bits)	Values	Comments
Message text	Variable	LV	Encapsulates the text of a message to user that is to be delivered to the final destination user.

6.7.4.4 SFO Filestore Request

One or more SFO Filestore Request messages may be optionally included and shall be constructed as indicated in table 6-28.

Table 6-28: SFO Filestore Request Message

Field	Size (bits)	Values	Comments
Length (octets)	8	Number of octets in the request field.	Encapsulates a filestore request that is to be executed by the local CFDP entity of the final destination user.
Request	8 × length	The value of this field is a single CFDP filestore request as defined in the CFDP Recommendation.	

6.7.4.5 SFO Fault Handler Override

One or more SFO Fault Handler Override messages may optionally be included and shall be constructed as indicated in table 6-29.

Table 6-29: SFO Fault Handler Override Message

Field	Size (bits)	Values	Comments
Fault handler override	8	As defined in the CFDP Recommendation.	Applies to each transmission to or by a waypoint.

6.7.4.6 SFO Flow Label

The SFO Flow Label message is optional and shall be constructed as indicated in table 6-30.

Table 6-30: SFO Flow Label Message

Field	Size (bits)	Values	Comments
Flow label	Variable	LV, as defined in the CFDP Recommendation.	Applies to each transmission to or by a waypoint.

6.7.5 RELAYING AN SFO TRANSMISSION FILE DATA UNIT

6.7.5.1 Upon receipt of a Transaction-Finished indication indicating successful reception of a file data unit whose metadata included an SFO Request message, the CFDP user shall invoke the SFO Routing procedure (see 6.7.3) to determine a route from its local CFDP entity to that of the final destination user.

6.7.5.2 If the computed route is an empty list, then relaying is impossible. The CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source user application of the routing failure.

6.7.5.3 If the computed route contains only the local CFDP entity, then the user application is the final destination of the file. No further SFO functionality shall apply.

6.7.5.4 If the computed route contains one or more waypoints in addition to the local CFDP entity and the final destination user's CFDP entity, then the user application shall use the CFDP Put.request primitive to request delivery of the received file (if any) and all received SFO Request, SFO Flow Label, SFO Fault Handler Override, SFO Message to User, and SFO Filestore Request messages to the first waypoint in the route. The transmission mode, segmentation control, fault handler overrides, and flow label that constrain the resulting transaction shall be those specified in the metadata of the received FDU.

If the transaction finishes in any condition other than 'No error', then the CFDP user shall use the SFO Reporting procedure (see 6.7.6) to notify the original source and final destination user applications of the relay transaction failure.

If the resulting transaction finishes in 'No error' condition, provided the Trace Requested flag in the SFO Request Message is set to 1, the user application shall use the SFO Reporting procedure (see 6.7.6) to report to the original source and final destination user applications on the success of the relay transaction.

NOTE – At any time after the transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.5.5 If the computed route contains only the local CFDP entity and that of the final destination user, then the user application—now serving as the agent of the SFO transmission operation—shall initiate an *SFO final delivery transaction*. To do so, it shall use the CFDP Put.request primitive to request delivery of a file delivery unit with the following parameters:

- destination entity ID as specified in the SFO Request message;
- source file name as specified in the SFO Request message (if any);
- destination file name as specified in the SFO Request message (if any);
- segmentation control as specified in the SFO Request message;
- transmission mode as specified in the SFO Request message;
- flow label as specified in the SFO Flow Label message that was received in the same transaction's Metadata-recv.indication (if any);
- one fault handler override for each SFO Fault Handler Override message that was received in the same transaction's Metadata-recv.indication;
- one Message to User for each SFO Message to User message that was received in the same transaction's Metadata-recv.indication;
- one filestore request for each SFO Filestore Request message that was received in the same transaction's Metadata-recv.indication;
- the entire SFO Request message, as an additional Message to User.

When the resulting transaction finishes, the user application shall use the SFO Reporting procedure (see 6.7.6) to report to the original source and final destination user applications on the success or failure of the final file delivery transaction.

NOTE – At any time after the transaction finishes, the user application may at its option delete its local copy of the transmitted file.

6.7.6 SFO REPORTING

6.7.6.1 General

6.7.6.1.1 When required to report on the results of an SFO file transmission, the user application shall invoke the SFO Routing procedure (see 6.7.3) to determine routes from its local CFDP entity to those of the original source user and final destination user.

6.7.6.1.2 If the computed route to the source is an empty list, then reporting to the source is impossible. No further SFO functionality shall apply.

6.7.6.1.3 If the computed route to the source contains only the local CFDP entity, then the user application is the original source of the file. Therefore reporting to the source is unnecessary, so no further SFO functionality shall apply.

6.7.6.1.4 If the computed route to the source contains more than one entity, then the user application shall use the CFDP Put.request primitive to request delivery of an *SFO transmission report* file delivery unit with the following parameters:

- destination entity ID set to the ID of the second entity in the computed route (the first one following the local CFDP entity);
- no file, source file name, or destination file name;
- a single SFO Report message, defined in 6.7.6.2;
- zero or more SFO Filestore Response messages, defined in 6.7.6.3, as applicable.

When the resulting transaction finishes, or if for any reason the Put.request cannot be honored, the user application shall take no further action with regard to this SFO transmission.

6.7.6.1.5 If the computed route to the destination is an empty list, then reporting to the destination is impossible. No further SFO functionality shall apply.

6.7.6.1.6 If the computed route to the destination contains only the local CFDP entity, then the user application is the final destination of the file. Therefore reporting to the destination is unnecessary, so no further SFO functionality shall apply.

6.7.6.1.7 If the computed route to the destination contains more than one entity, then the user application shall use the CFDP Put.request primitive to request delivery of an *SFO transmission report* file delivery unit with the following parameters:

- destination entity ID set to the ID of the second entity in the computed route (the first one following the local CFDP entity);
- no file, source file name, or destination file name;
- a single SFO Report message, defined in 6.7.6.2;
- zero or more SFO Filestore Response messages, defined in 6.7.6.3, as applicable.

When the resulting transaction finishes, or if for any reason the Put.request cannot be honored, the user application shall take no further action with regard to this SFO transmission.

6.7.6.2 SFO Report

The SFO Report message is mandatory and shall be constructed as indicated in table 6-31. If the SFO transmission report is the result of completion of an SFO final delivery transaction, then the message's condition code, delivery code, and file status shall be obtained from the Transaction-Finished.indication that signaled that completion; otherwise the values of all three fields shall be undefined.

Table 6-31: SFO Report Message

Field	Size (bits)	Values	Comment
SFO request label	Variable	LV	Copied from the request label in the SFO Request message for this SFO transmission.
Source entity ID	Variable	LV	The ID of the original source user's CFDP entity for this SFO transmission.
Destination entity ID	Variable	LV	The ID of the final destination user's CFDP entity for this SFO transmission.
Reporting entity ID	Variable	LV	The ID of the CFDP entity of the user which originally issued this SFO transmission report.
Prior waypoints count	8		Copied from the prior waypoints count in the SFO Request message for this SFO transmission.

Field	Size (bits)	Values	Comment
Report code	8	1, 2, 3, or 4	1: final file delivery transaction completed. 2: relay transaction failed. 3: routing failed, relay was not possible. 4: relay transaction succeeded.
Condition code	4	See the CFDP Recommendation.	
Direction	1	0: report is destined for the original source user application. 1: report is destined for the final destination user application.	
Delivery Code	1	See the CFDP Recommendation.	
File Status	2	See the CFDP Recommendation.	

6.7.6.3 SFO Filestore Response

If the SFO transmission report is the result of completion of an SFO final delivery transaction, and if the Transaction-Finished indication signaling this transaction completion included one or more filestore responses, then for each such response one SFO Filestore Response message shall be included. Each such message shall be constructed as indicated in table 6-12.

6.7.7 RELAYING AN SFO TRANSMISSION REPORT

6.7.7.1 Upon receipt of a Transaction-Finished indication indicating successful reception of a file data unit whose metadata included an SFO Report message, the CFDP user shall invoke the SFO Routing procedure (see 6.7.3) to determine a route from its local CFDP entity to that of the user application to which the report is directed (as indicated by the Direction noted in the SFO Report message).

6.7.7.2 If the computed route is an empty list, then relaying is impossible. The user application shall take no further action with regard to this SFO transmission.

6.7.7.3 If the computed route contains only the local CFDP entity, then the user application is the original source of the file. No further SFO functionality shall apply.

6.7.7.4 If the computed route contains more than one entity, then the user application shall use the CFDP Put request primitive to request delivery of the received SFO Report and SFO Filestore Response message(s) to the second entity in the computed route (the first one following the local CFDP entity).

7 CFDP SERVICE CLASSES

NOTE – This section describes the minimum functionality of the CFDP protocol required to realize interoperable implementations of certain identified application services. The classes are, as yet, neither definitive nor complete. For each class, an event diagram is included to illustrate the sequence of events associated with the use of that protocol class.

7.1 DEFINED CLASSES

The defined CFDP protocol classes are:

- a) Class 1—Unreliable Transfer;
- b) Class 2—Reliable Transfer;
- c) Class 3—Unreliable Transfer ~~By Proxy~~ Via One Waypoint;
- d) Class 4—Reliable Transfer Via One Waypoint.

NOTES

- 1 State Transition Diagrams and State Tables for each of the Service Classes are provided in reference [A4].
- 2 The following subsections describe the functions and procedures, contained in sections ~~4 and 6~~ of this document, that are applicable to each Class. In order to implement the Class, it is only necessary to make reference to the General Characteristics overview of 2.3, the subsections specified in the Class procedures, and any PDU descriptions (section 5) referenced in the procedures.

7.4 FUNCTIONS OF CLASS 3—UNRELIABLE TRANSFER VIA ONE WAYPOINT

NOTE — Class 3 provides for the unreliable delivery of bounded or unbounded data files from a source to the destination through the mediation of a waypoint.

7.4.1 PROCEDURES FOR CLASS 3—SOURCE

For Class 3 functionality, the source CFDP entity shall use the procedures specified in table 7-5.

Table 7-5: Class 3 Source Procedures

<u>Procedure</u>	<u>Subsection</u>	<u>Notes</u>
<u>Put Procedures</u>	<u>4.1.3</u>	
<u>Transaction Start Notification Procedure</u>	<u>4.1.4</u>	
<u>PDU Forwarding Procedures</u>	<u>4.1.5</u>	<u>Applicable to Metadata, File Data and EOF PDUs only.</u>
<u>Copy File Procedures at the Sending Entity</u>	<u>4.1.6.1.1</u>	
<u>Unacknowledged Mode Procedures at the Sending Entity</u>	<u>4.1.6.3.1</u>	
<u>Fault Handling Procedures</u>	<u>4.1.8</u>	<u>Only issuance of Notice of Cancellation is required.</u>
<u>Notice of Completion Procedures at the Sending Entity</u>	<u>4.1.11.1.1</u>	
<u>Notice of Cancellation Procedures at the Sending Entity</u>	<u>4.1.11.2.2</u>	<u>Positive acknowledgement not required.</u>

7.4.2 PROCEDURES FOR CLASS 3—WAYPOINT

For Class 3 functionality, the waypoint CFDP entity shall use the procedures specified in table 7-6.

Table 7-6: Class 3 Waypoint Procedures

Procedure	Subsection	Notes
PDU Forwarding Procedures	4.1.5	Applicable to Metadata, File Data and EOF PDUs only.
Copy File Procedures at the Sending Entity	4.1.6.1.1	
Copy File Procedures at the Receiving Entity	4.1.6.1.2	
Unacknowledged Mode Procedures at the Sending Entity	4.1.6.3.1	
Unacknowledged Mode Procedures at the Receiving Entity	4.1.6.3.2	
Cancel Response Procedures at the Receiving Entity	4.1.6.6.1	Positive acknowledgement not required.
Fault Handling Procedures	4.1.8	Abandonment of transaction only required.
Notice of Cancellation Procedures	4.1.11.2	Positive acknowledgement not required.

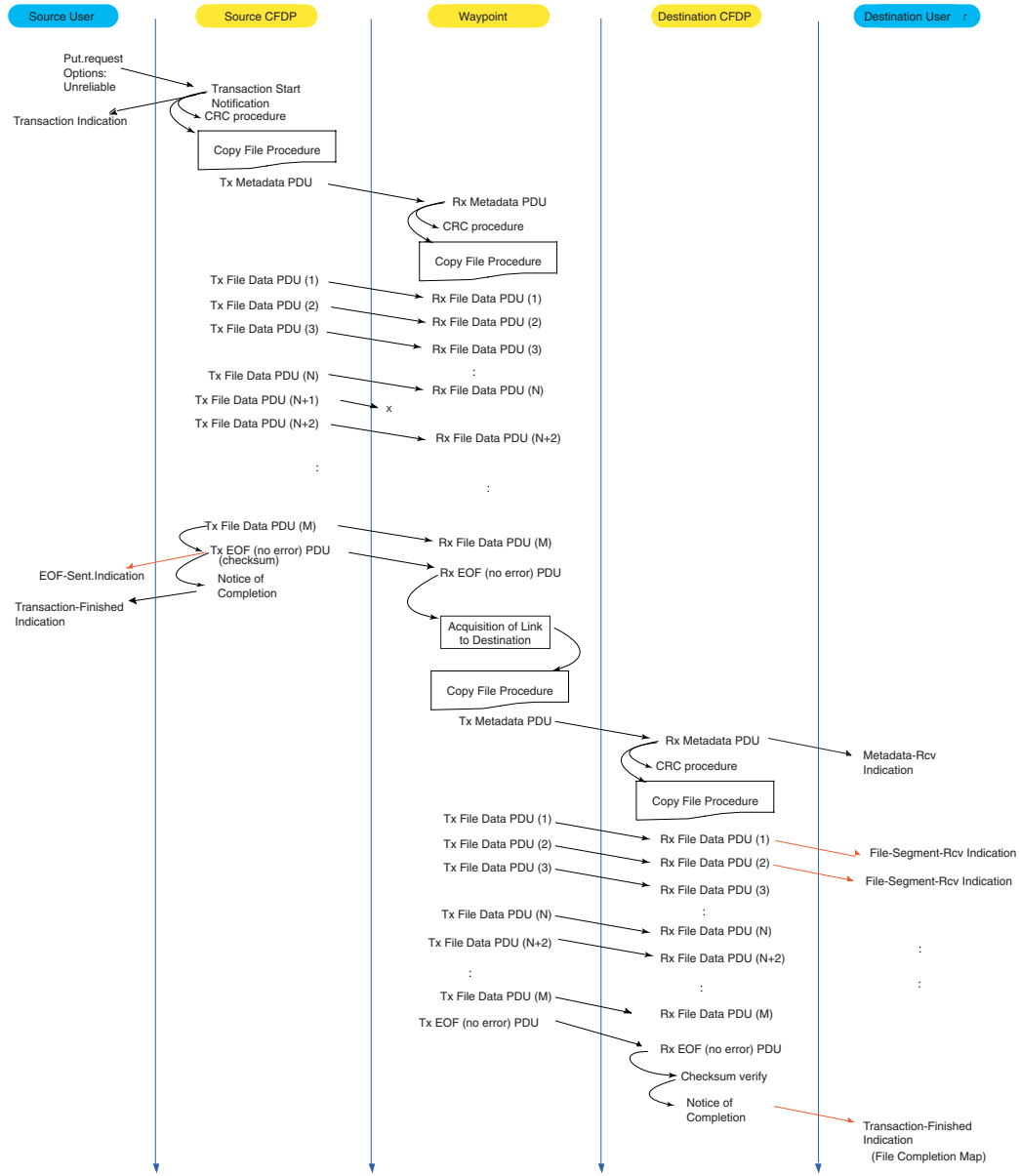
7.4.3 PROCEDURES FOR CLASS 3—DESTINATION

For Class 3 functionality, the destination CFDP entity shall use the procedures specified in table 7-7.

Table 7-7: Class 3 Destination Procedures

Procedure	Subsection	Notes
Copy File Procedures at the Receiving Entity	4.1.6.1.2	
Unacknowledged Mode Procedures at the Receiving Entity	4.1.6.3.2	
Cancel Response Procedures at the Receiving Entity	4.1.6.6.1	Positive acknowledgement not required.
Fault Handling Procedures	4.1.8	Abandonment of transaction only required.
Notice of Completion Procedures at the Receiving Entity	4.1.11.1.2	

7.4.4 EVENT DIAGRAM FOR CLASS 3



NOTE – Waypoint use of the Suspend/Resume Procedures and the Copy File Procedures at the Sending Entity entails potential interaction with a User Application. When a CFDP entity is operating as a Waypoint, the entity itself functions as the User Application; therefore for the sake of clarity in this diagram, interaction with the User Application is not shown. The potential interaction is limited to EOF-Sent.indication, Suspend.indication, and Resume.request as illustrated in the diagram for Service Class 2.

7.5 FUNCTIONS OF CLASS 4—RELIABLE TRANSFER VIA ONE WAYPOINT

NOTE – Class 4 provides for the reliable delivery of bounded or unbounded data files from a source to the destination through the mediation of a waypoint.

7.5.1 PROCEDURES FOR CLASS 4—SOURCE

For Class 4 functionality, the source CFDP entity shall use the procedures specified in table 7-8.

Table 7-8: Class 4 Source Procedures

Procedure	Subsection	Notes
CRC Procedures at the PDU Transmitting Entity	4.1.1.1	
CRC Procedures at the PDU Receiving Entity	4.1.1.2	
Put Procedures	4.1.3	
Transaction Start Notification Procedure	4.1.4	
PDU Forwarding Procedures	4.1.5	
Copy File Procedures at the Sending Entity	4.1.6.1.1	
Acknowledged Mode Procedures at the Sending Entity	4.1.6.4.1	
Cancel Response Procedures at the Sending Entity	4.1.6.6.2	
Resume Procedures	4.1.6.7	Only procedures applicable to the sending entity.
Positive Acknowledgement Procedures at PDU Sending End	4.1.7.1	
Positive Acknowledgement Procedures at PDU Receiving End	4.1.7.2	
Fault Handling Procedures	4.1.8	Only issuance of Notice of Cancellation is required.
Internal Procedures	4.1.11	Positive acknowledgement required. Only procedures applicable to the sending entity are required.

7.5.2 PROCEDURES FOR CLASS 4—WAYPOINT

For Class 4 functionality, the waypoint CFDP entity shall use the procedures specified in table 7-9.

Table 7-9: Class 4 Waypoint Procedures

Procedure	Subsection	Notes
CRC Procedures at the PDU Transmitting Entity	4.1.1.1	
CRC Procedures at the PDU Receiving Entity	4.1.1.2	
PDU Forwarding Procedures	4.1.5	
Copy File Procedures at the Sending Entity	4.1.6.1.1	
Copy File Procedures at the Receiving Entity	4.1.6.1.2	
Acknowledged Mode Procedures at the Sending Entity	4.1.6.4.1	
Acknowledged Mode Procedures at the Receiving Entity	4.1.6.4.2	
Incremental Lost Segment Detection Procedures at the Receiving Entity	4.1.6.4.3	Deferred Mode required only for interoperability.
Cancel Response Procedures	4.1.6.6	Positive acknowledgement required. Incomplete data forwarded.
Resume Procedures	4.1.6.7	
Positive Acknowledgement Procedures at PDU Sending End	4.1.7.1	
Positive Acknowledgement Procedures at PDU Receiving End	4.1.7.2	
Fault Handling Procedures	4.1.8	Only issuance of Notice of Cancellation is required.
Internal Procedures	4.1.11	
Finished (no error) PDU Relay Procedure	4.2.3	
Cancel Propagation Procedures	4.2.4	
Deferred Transmission Procedure	4.2.5	

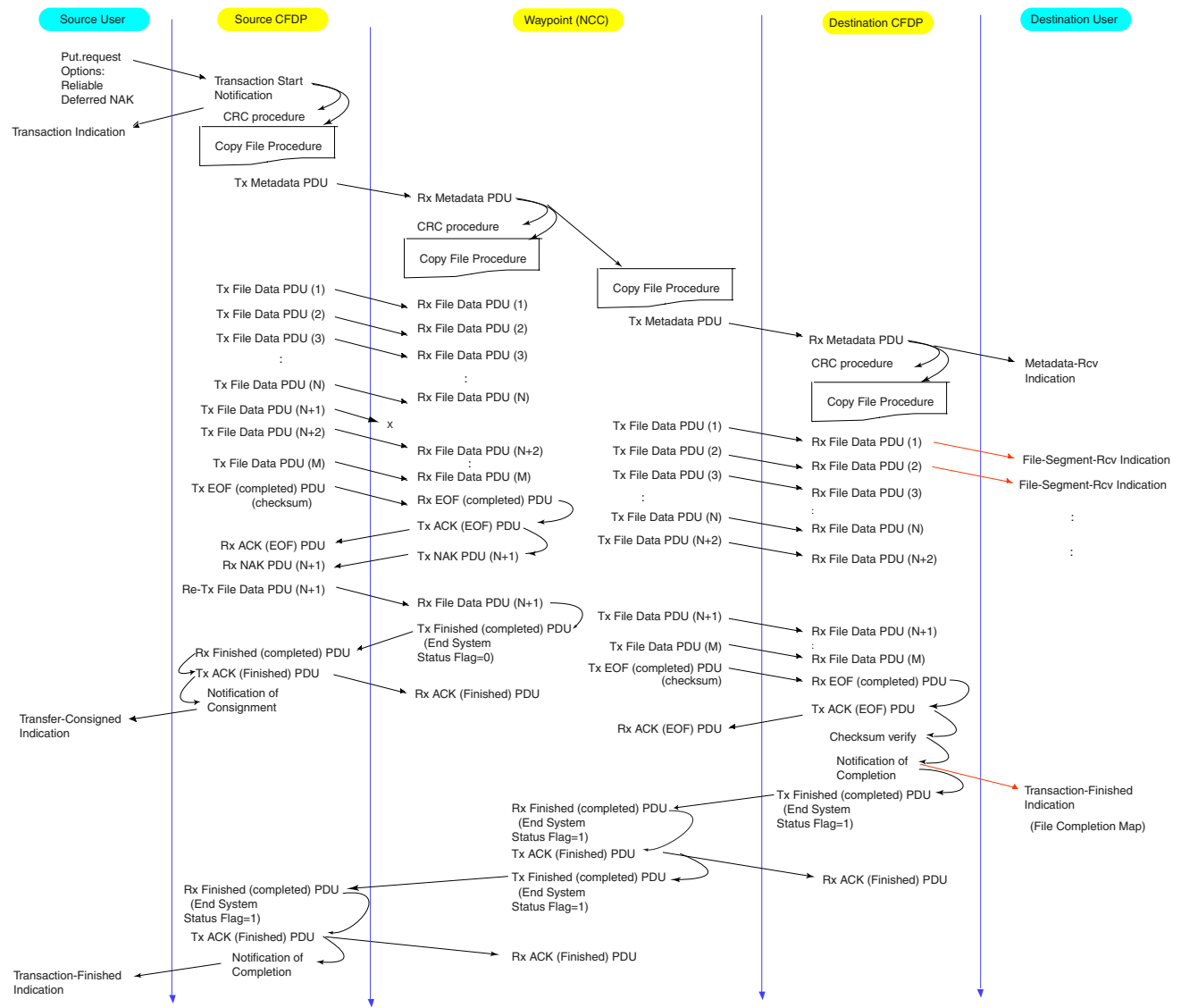
7.5.3 PROCEDURES FOR CLASS 4—DESTINATION

For Class 4 functionality, the destination entity shall use the procedures specified in table 7-10.

Table 7-10: Class 4 Destination Procedures

Procedure	Subsection	Notes
CRC Procedures at the PDU Transmitting Entity	4.1.1.1	
CRC Procedures at the PDU Receiving Entity	4.1.1.2	
PDU Forwarding Procedures	4.1.5	
Copy File Procedures at the Receiving Entity	4.1.6.1.2	
Acknowledged Mode Procedures at the Receiving Entity	4.1.6.4.2	
Incremental Lost Segment Detection Procedures at the Receiving Entity	4.1.6.4.3	Deferred Mode only required for interoperability.
Cancel Response Procedures at the Receiving Entity	4.1.6.6.1	Positive acknowledgement required. Incomplete data forwarded.
Resume Procedures	4.1.6.7	Only procedures applicable to the receiving entity are required.
Positive Acknowledgement Procedures at PDU Sending End	4.1.7.1	
Positive Acknowledgement Procedures at PDU Receiving End	4.1.7.2	
Fault Handling Procedures	4.1.8	Only issuance of Notice of Cancellation is required.
Internal Procedures	4.1.11	Incomplete data forwarded. Only procedures applicable to the receiving entity are required.
Deferred Transmission Procedure	4.2.5	

7.5.4 EVENT DIAGRAM FOR CLASS 4



NOTE – Waypoint use of the Suspend/Resume Procedures and the Copy File Procedures at the Sending Entity entails potential interaction with a User Application. When a CFDP entity is operating as a Waypoint, the entity itself functions as the User Application; therefore for the sake of clarity in this diagram, interaction with the User Application is not shown. The potential interaction is limited to EOF-Sent.indication, Suspend.indication, and Resume.request as illustrated in the diagram for Service Class 2.

8 MANAGEMENT INFORMATION BASE

8.1 GENERAL

The operation of each CFDP entity shall be supported by a single Management Information Base (MIB) comprising the items of information described below.

NOTES

- 1 This section describes the Management Information Base.
- 2 Representation of, and mechanisms for access to, MIB items will be implementation matters. In particular, determination of which items will be static and which will be dynamic is a matter of implementation.

8.2 LOCAL ENTITY CONFIGURATION INFORMATION

For each item of local entity configuration information, a single value shall apply which shall pertain to the entire CFDP entity.

Table 8-1: Local Entity Configuration Information

Item	Comment
Local Entity ID	
EOF-Sent.indication required	True or false.
File-Segment-Recv.indication required	True or false.
Transaction-Finished.indication required when acting as receiving entity	True or false.
Suspended.indication required when acting as receiving entity	True or false.
Resumed.indication required when acting as receiving entity	True or false.
Default fault handlers	For each type of fault condition, a default handler (as enumerated in 4.1.8.2).
Routing information (Extended procedures only)	For each entity in the system, a list of all adjacent entities.

~~ANNEX D~~

~~EXTENDED PROCEDURES~~

[This annex has been deleted.]